SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Courier Management System for LPU

Submitted By

Name	Registration No.	Roll NO
• Abhishek kumar	12105161	RK21QAA14
• Prashant Thakur	12105797	RK21QAB39
•Sachin Vishwakarma	12107934	RK21QAB63

Submitted to

NIDHI ARORA

Introduction

CMS is a web-based Courier Management System that supports the high accessibility of courier services to the corporate and to the customer. The system is being used for day-to-day activities such as booking a courier, tracking courier and many other things.

CMS can be personalized to fit your business and can either be used as a complete system or as separate modules. This idea of the project represents the 'Courier Service Management System'. The system is being used for day-to-day actions such as maintain employee details, booking a courier, maintain corporation details, process data of employees, and many other things.

Courier management computerization is "the incorporation of appropriate technology to help managers manage information. Technology is considered suitable when it utilizes the most abundant domestic possessions and conserves investment and skilled personnel". The main aim of this project is to computerize the maintenance of courier management.

Objectives and Success Criteria of Courier Management System for LPU

The objective of the project is to deliver an efficient Courier Management System in campus whose main functionality apart from calculating the courier Bill includes predicting the time required to reach the destination. As per our client requirement, our main object should be managing consignment in an effective manner.

The Proposed System is eliminating all problems of the existing system. And automate all processes in high-tech.

It should keep a record of customer booking and delivery details and so on can be proficient without much effort the success criteria depend on:

- The accuracy in calculating the bill for each consignment.
- The accuracy in predicting the time required to reach the destination.
- User interface simplicity and user-friendliness.

Purpose of Courier Management System

The system will be used for day-to-day activities like out return, company details, hub rates, booking, non-delivery, and pickup centers. It is not easy to do this process manually because it would become very hectic. Hence it is suggested to automate the process by developing the relevant software as the world is moving from manual working to an information and technology era where automation becomes important in all parts of life.

The main purpose of this system is to connect all branches to the central database so the everywhere information is the same. This system increases efficiency and increases the customer satisfaction level.

User Interface Design of Courier Management System

Interface plays a major role in the success of any software. Simple, user-friendly, and standard UI makes visitors' experience great and which means the software is going to be successful. Sometimes very smooth and has very good database design software can't able to make their existence in the market because of its Interface .

The Courier Management System hassles out all the complications of the conventional method which is a combination of the interaction to human. Each profile has its own UI which is customized by our experienced team to make them feel like they are special.

Code used in making CMS

```
from tkinter import *
from tkinter import messagebox as ms
from tkinter import ttk
import sqlite3
import random
from tkinter import Button


# Database



with sqlite3.connect('LPU.db') as db:
    c = db.cursor()
try:
    c.execute('CREATE TABLE IF NOT EXISTS user (username
TEXT NOT NULL ,password TEX  NOT NULL,mobile TEX NOT
NULL);')
except:
    pass
db.commit()
db.close()



class main:
    def __init__(self,master):


        self.master = master
```

```
        self.username = StringVar()
        self.password = StringVar()
        self.n_username = StringVar()
        self.n_password = StringVar()
        self.n_reg=StringVar()
        self.n_mobile=StringVar()
        self.mobile11=StringVar()
        self.widgets()




    def login(self):

        with sqlite3.connect('LPU.db') as db:
            c = db.cursor()


        #Find user If there is any take proper action
        find_user = ('SELECT * FROM user WHERE username =
? and password = ?')
        c.execute(find_user,[(self.username.get()),(self.p
assword.get())])
        result = c.fetchall()


        if result:
            self.track()
        else:
            ms.showerror('Oops!','Username Not Found.')

    def new_user(self):



        with sqlite3.connect('LPU.db') as db:
            c = db.cursor()
        if self.n_username.get()!=' ' and
self.n_password.get()!=' ' and self.n_mobile.get()!=' ':
```

```
            find_user = ('SELECT * FROM user WHERE
username = ?')
            c.execute(find_user,[(self.n_username.get())])



            if c.fetchall():
                    ms.showerror('Error!','Username Taken
Try a Diffrent One.')
            else:
                    insert = 'INSERT INTO
user(username,password,mobile) VALUES(?,?,?)'
                    c.execute(insert,[(self.n_username.get
()),(self.n_password.get()),(self.n_mobile.get())])
                    db.commit()

                    ms.showinfo('Success!','Account
Created!')
                    self.log()
        else:
             ms.showerror('Error!','Please Enter the
details.')



    def consignment(self):



        try:
            with sqlite3.connect('LPU.db') as db:
                c = db.cursor()


            #Find user If there is any take proper action
            find_user = ('SELECT * FROM user WHERE mobile=
?')
            c.execute(find_user,[(self.mobile11.get())])
            result = c.fetchall()

            if result:
                    self.track()
```

```
                    self.crff.pack_forget()
                    self.head['text'] =
self.username.get() + '\n Your Product Details'
                    self.consi.pack()
            else:
                    ms.showerror('Oops!','Mobile Number
Not Found.')
        except:
            ms.showerror('Oops!','Mobile Number Not
Found.')




    def track1(self):
        self.consi.pack_forget()
        self.head['text'] = self.username.get() + '\n
Track your Product'
        self.crff.pack()
    def log(self):
        self.username.set('')
        self.password.set('')
        self.crf.pack_forget()
        self.head['text'] = 'Login'
        self.logf.pack()


    def cr(self):
        self.n_username.set('')
        self.n_password.set('')
        self.logf.pack_forget()
        self.head['text'] = 'Create Account'
        self.crf.pack()


    def track(self):
            self.logf.pack_forget()
            self.head['text'] = self.username.get() + '\n
Track your Product'

            self.crff.pack()
```

```
    def widgets(self):
        self.head = Label(self.master,text = 'LOGIN',font
= ('',20),pady = 10)
        self.head.pack()



        self.logf = Frame(self.master,padx =10,pady = 10)


        self.logf.configure(background='lightblue')
        #PhotoImage(self.logf,file = 'lpu_logo.png')
        Label(self.logf,text = 'Username: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.logf,textvariable = self.username,bd =
3,font = ('',15)).grid(row=0,column=1)
        Label(self.logf,text = 'Password: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.logf,textvariable = self.password,bd =
3,font = ('',15),show = '*').grid(row=1,column=1)
        Button(self.logf,text = ' Login
',background='lightgrey',bd = 2 ,font =
('',13),padx=6,pady=6,command=self.login).grid(row=8,colum
n=0)
        Button(self.logf,text = ' New user
',background='lightgrey',bd = 2 ,font =
('',13),padx=6,pady=6,command=self.cr).grid(row=8,column=1
)



        self.logf.pack()

        self.crf = Frame(self.master,padx =10,pady = 10)
        Label(self.crf,text = 'Username: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.crf,textvariable = self.n_username,bd =
3,font = ('',15)).grid(row=0,column=1)
```

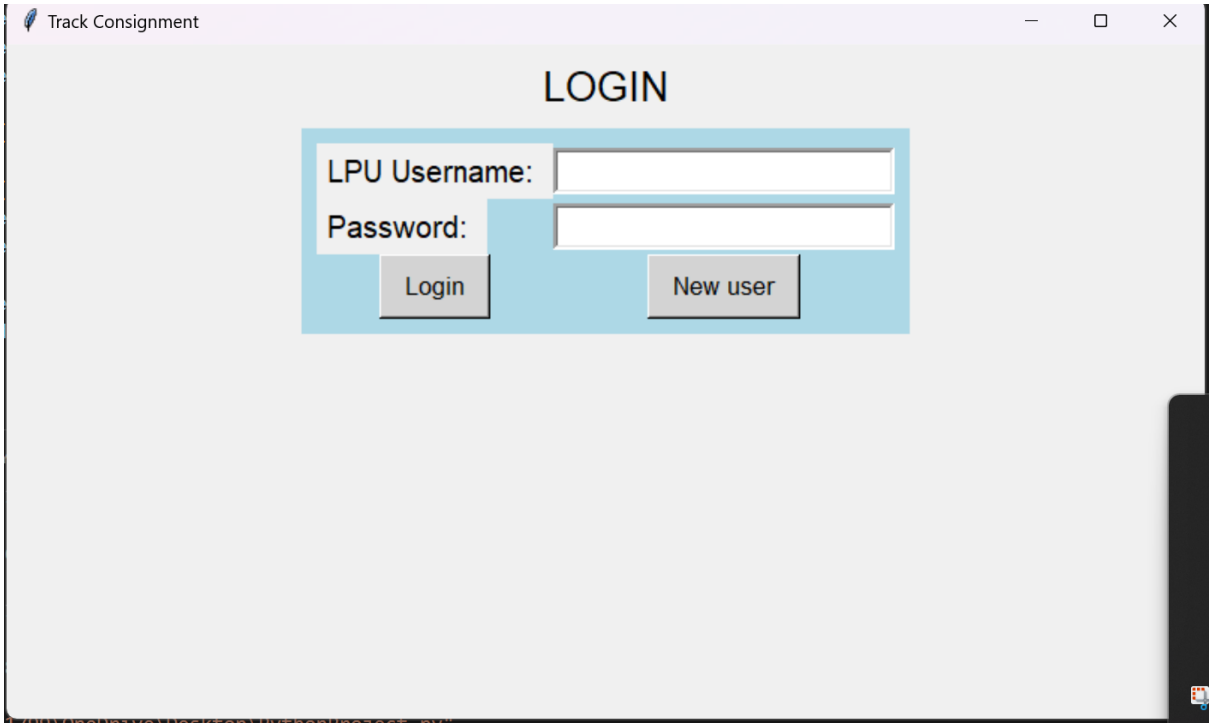
```
        Label(self.crf,text = 'Password: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.crf,textvariable = self.n_password,bd =
3,font = ('',15),show = '*').grid(row=1,column=1)

        Label(self.crf,text = 'Reg No.: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.crf,textvariable = self.n_reg,bd =
3,font = ('',15)).grid(row=2,column=1)
        Label(self.crf,text = 'Gender: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        var = IntVar()
        R1 = Radiobutton(self.crf, text="Male",
variable=var, value=1).grid(sticky=W)


        R2 = Radiobutton(self.crf, text="Female",
variable=var, value=2 ).grid(row=4,column=1)
        Label(self.crf,text = 'Mobile No.: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.crf,textvariable = self.n_mobile,bd =
3,font = ('',15)).grid(row=5,column=1)


        Label(self.crf,text = 'Email Id: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.crf,bd = 3,font =
('',15)).grid(row=6,column=1)




        Button(self.crf,text = 'Create
Account',background='lightgrey',bd = 2,font =
('',13),padx=6,pady=6,command=self.new_user).grid(row=11,c
olumn=0)
        Button(self.crf,text = 'Go to
Login',background='lightgrey',bd = 2,font =
('',13),padx=6,pady=6,command=self.log).grid(row=11,column
=1)
```

```
        self.crff = Frame(self.master,padx =10,pady = 10)



        Label(self.crff,text = 'Consignment No: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.crff,bd = 3,font =
('',15)).grid(row=0,column=1)
        Label(self.crff,text = 'Mobile no:',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Entry(self.crff,bd = 3,textvariable =
self.mobile11,font = ('',15)).grid(row=1,column=1)
        Button(self.crff,text =
'Track',background='lightgrey',bd = 2,font =
('',13),padx=6,pady=6,command=self.consignment).grid(row=4
,column=0)



        self.consi = Frame(self.master,padx =10,pady = 10)



        Label(self.consi,text = ' Product ID:',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Label(self.consi,text
=random.randint(565154,99994216) ,font =
('',13),pady=5,padx=5).grid(row=0,column=1)
        L = ['Bag','Colgate','shoe','Redme
2','Jeans','Parrot','Mac','Ipad','Pen','Book','shirt']
        f=random.randint(0,10)
        Label(self.consi,text = 'Product name: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Label(self.consi,text =L[f] ,font =
('',13),pady=5,padx=5).grid(row=1,column=1)
        Label(self.consi,text = 'Product Status: ',font =
('',15),pady=5,padx=5).grid(sticky = W)
        Label(self.consi,text ='Pending' ,font =
('',13),pady=5,padx=5).grid(row=2,column=1)
        Label(self.consi,font = ('',13), text = 'Thanks
for Exploring!').grid(row = 4, column = 0)
```

```
        Label(self.consi, text = 'Comments:',font =
('',13)).grid(row = 5, column = 0, padx = 5, sticky =
'sw')
        Entry(self.consi,bd = 3,font =
('',15)).grid(row=5,column=1)


        Button(self.consi,text =
'Back',background='lightgrey',bd = 2,font =
('',13),padx=6,pady=6,command=self.track1).grid(row=6,colu
mn=0)




if __name__ == '__main__':

    root = Tk()
    root.title('Track Consignment')
    root.geometry('800x450+300+300')
    main(root)

    root.mainloop()
```

Result

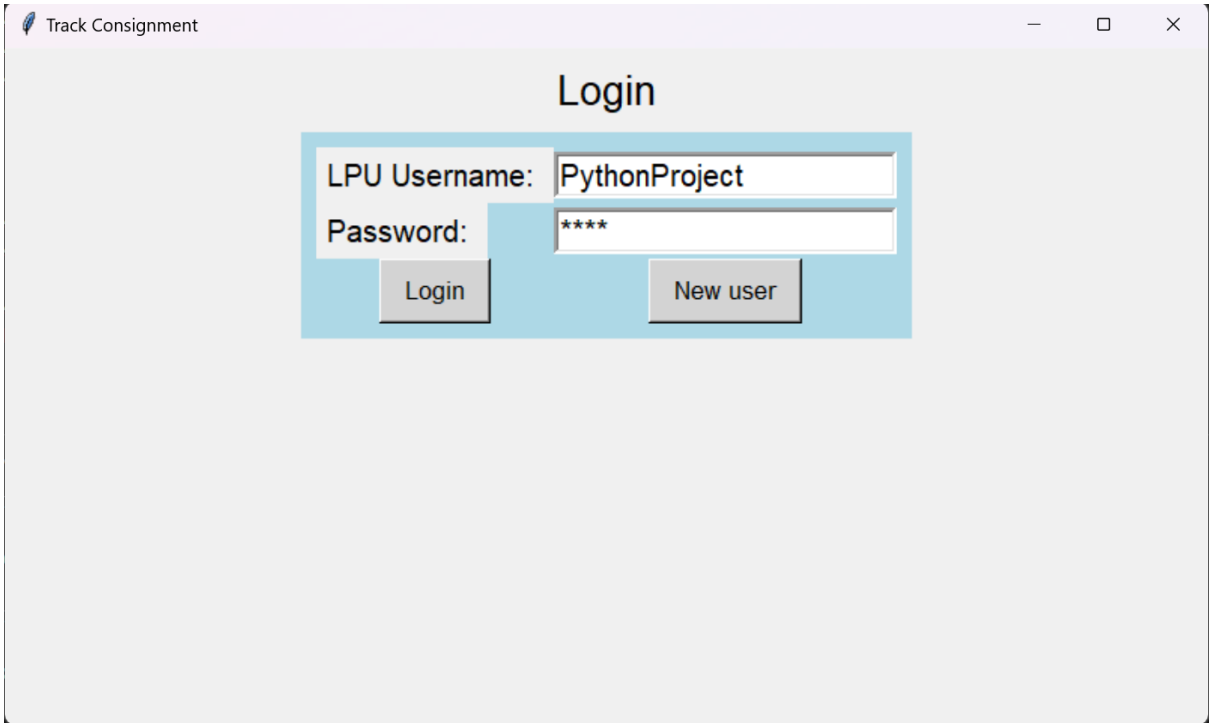
Login Page



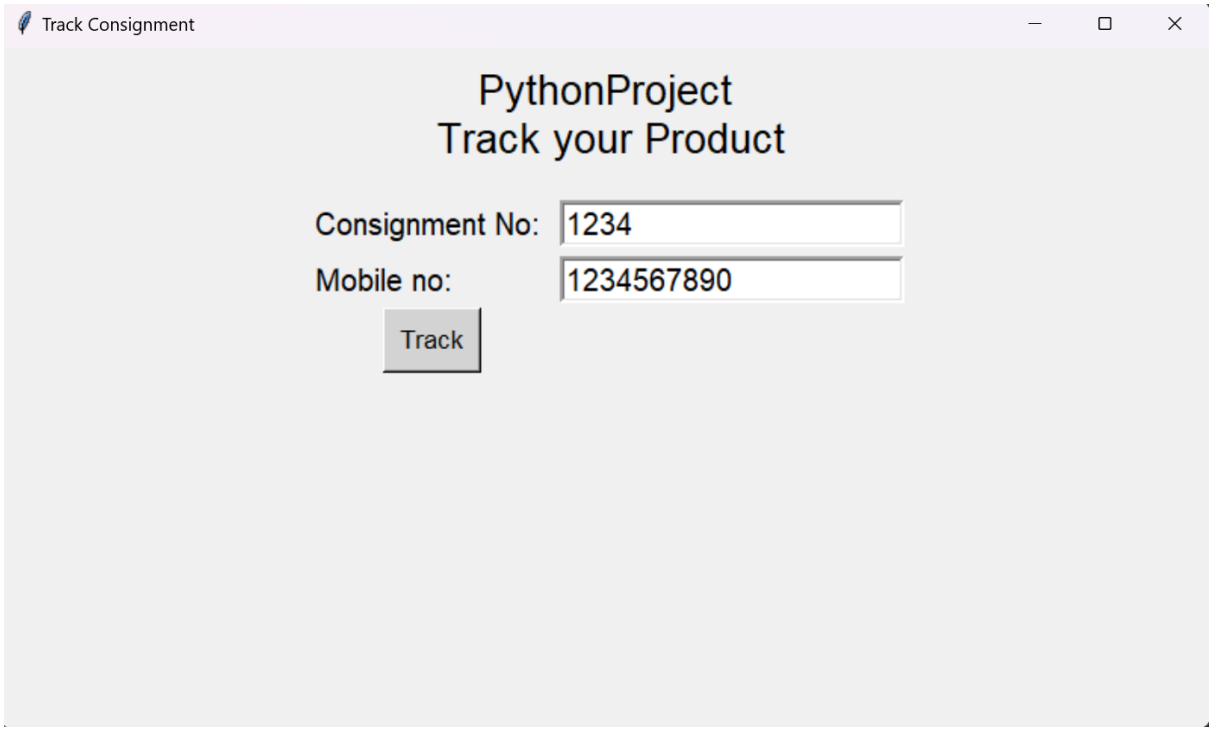
Basically, for any application security is a major concern. So, we have developed a secure application. Without being authenticated no user is allowed to view any other interfaces. For the login page, we have a User ID, Password . After being authenticated user is authorized to login .

The login page allows a user to gain access to an application by entering their username and password

Case1

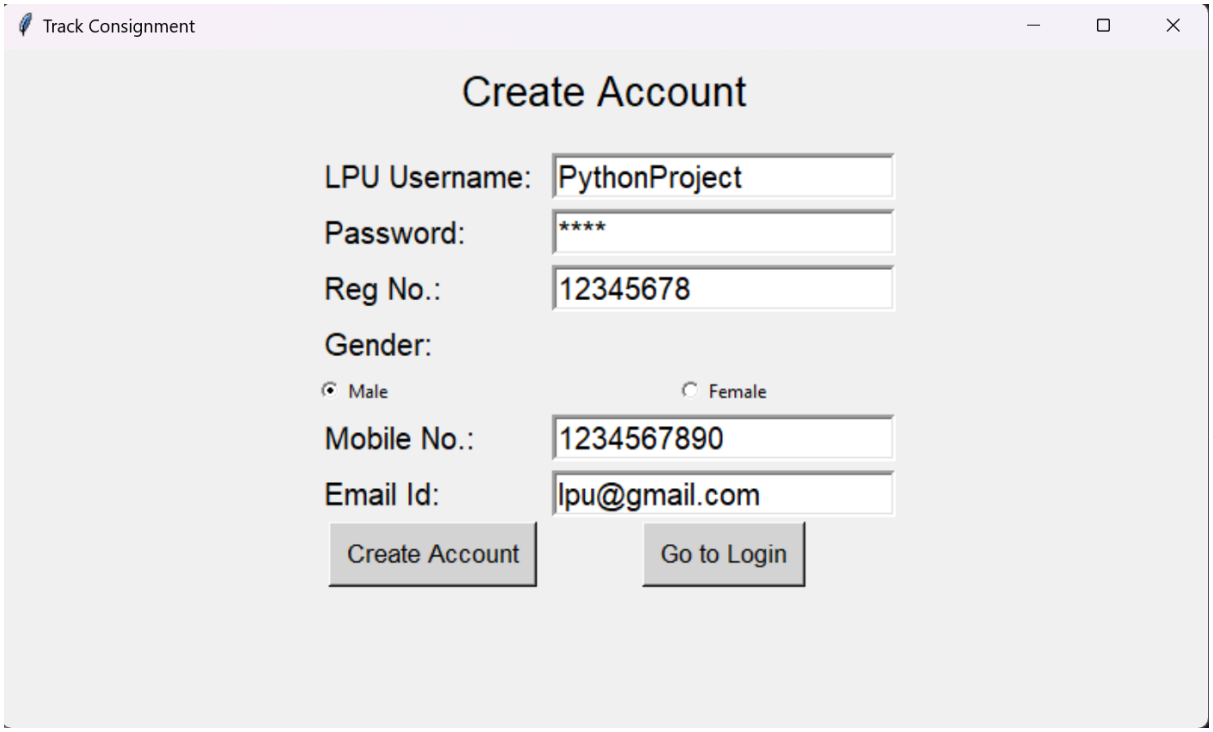


If the user enter the correct login credentials the user will be redirected to courier tracking page.



Registration page

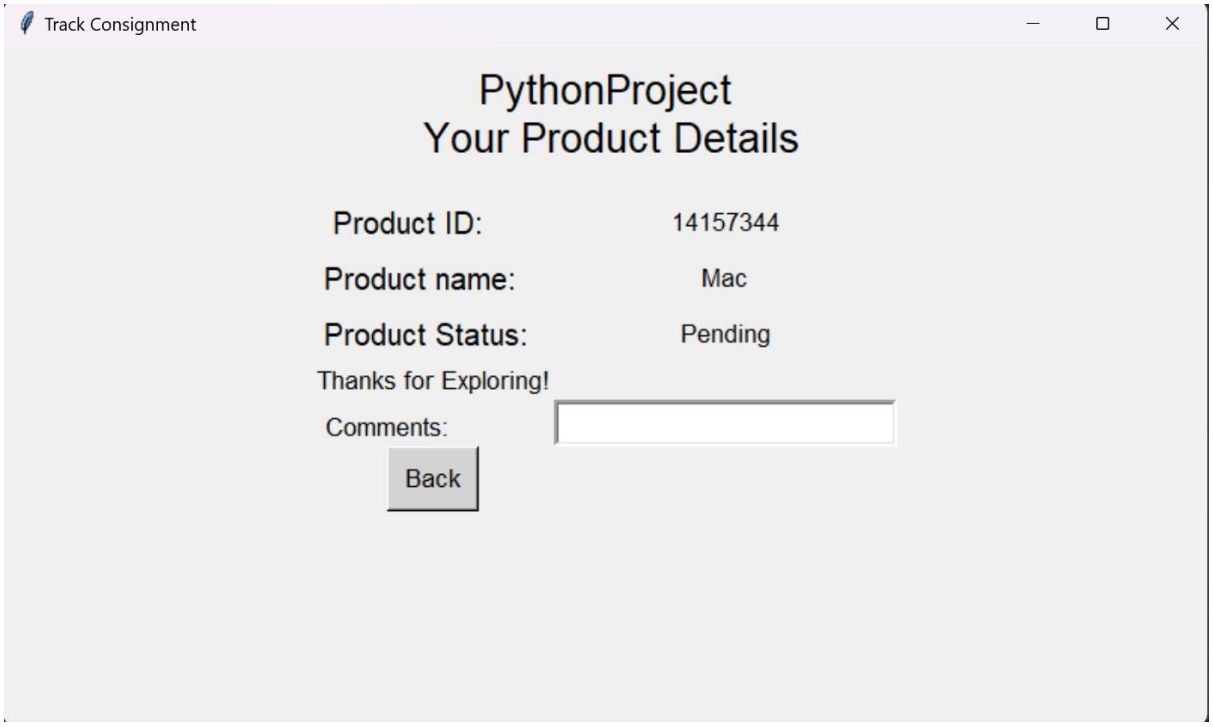
If the user doesn’t have the account they can create a new account by clicking the new user button on the login page.



Now user have to enter the all the details to create the new account, and the new account will be created .

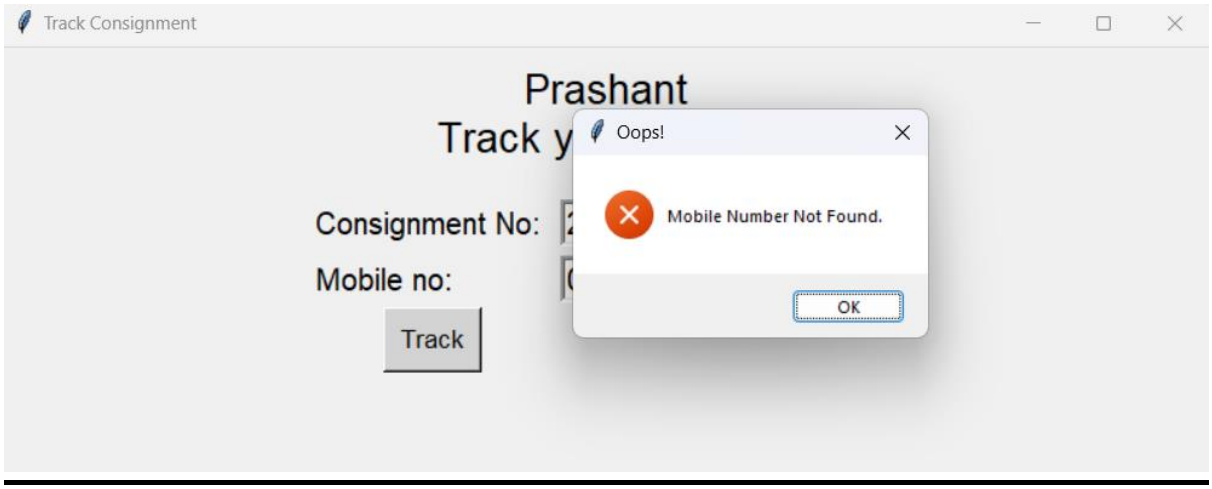
Tracking page

The user has to enter the consignment number and registered mobile number to see the product (Product ID, and Product name)



Case1:

If the user does not enter the correct registration number, then the user will redirected to the following page.



Conclusion of Courier Management System

The project titled ‘Courier Management Service’ was developed to the courier services and direction and with their help. The system was tested and the performance of the system was found to be acceptable. All the necessary output was created. The system was found to be user-friendly with a help message for the customer.